AD-751 542

A NEW FALLOUT TRANSPORT CODE FOR THE DELFIC SYSTEM: THE DIFFUSIVE TRANSPORT MODULE. SUPPLEMENT

Hillyer G. Norment

Mount Auburn Research Associates, Incorporated

Prepared for:

Defense Nuclear Agency

October 1972

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

AD 751542

DNA 2669
Supplement

October 1972

A NEW FALLOUT TRANSPORT CODE FOR THE DELFIC SYSTEM: THE DIFFUSIVE TRANSPORT MODULE - SUPPLEMENT

Prepared by

H. G. Norment

MT. AUBURN RESEARCH ASSOCIATES, INC.
385 Elliot Street
Newton, Massachusetts 02164

This work was supported by the Defense Nuclear Agency under NWED/Subtask PD068-04

Sponsored by

Defense Nuclear Agency
Washington, D. C. 20305

Contract No. DASA01-71-C-0118

Approved for public release; distribution unlimited

Destroy this report when it is no longer needed. Do not return to sender.

ACCESSION for

NTIS White Section ☒
DDC Buff Section ☐
UNANNOUNCED ☐
JUSTIFICATION

BY
DISTRIBUTION/AVAILABILITY CODES

Dist.	AVAIL. and/or SPECIAL	
A		

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) MT. AUBURN RESEARCH ASSOCIATES, INC. 385 Elliot Street Newton, Massachusetts 02164	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A New Fallout Transport Code for the DELFIC System: The Diffusive Transport Module - Supplement

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report, 6 May 1971 - 6 May 1972. Approved 3 October 1972.

5. AUTHOR(S) (First name, middle initial, last name)

Hillyer G. Norment

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1972	~~46~~ 43	0

8a. CONTRACT OR GRANT NO. DASA01-71-C-0118 b. PROJECT NO. NWED:XAXP c. Task & Subtask: D068 d. Work Unit: 04	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) DASA 2669 Supplement

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Defense Nuclear Agency Washington, D. C. 20305

13. ABSTRACT

The Diffusive Transport Module of the new DELFIC fallout prediction system has undergone additional development since publication of its description in DASA 2669. This supplement to DASA 2669 describes these developments and presents amendments and corrections to the code and its documentation. Complete FORTRAN statement listings of subroutines that have been substantially changed are included.

DD FORM 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Nuclear weapons effects						
Radioactivity						
Fallout prediction						
DELFIC						

unclassified

DNA 2669
Supplement

October 1972

A NEW FALLOUT TRANSPORT CODE FOR THE DELFIC SYSTEM: THE DIFFUSIVE TRANSPORT MODULE - SUPPLEMENT

Prepared by

H. G. Norment

MT. AUBURN RESEARCH ASSOCIATES, INC.
385 Elliot Street
Newton, Massachusetts 02164

This work was supported by the Defense Nuclear Agency
under NWED/Subtask PD068-04

Sponsored by

Defense Nuclear Agency
Washington, D. C. 20305

Contract No. DASA01-71-C-0118

Approved for public release; distribution unlimited

ABSTRACT

The Diffusive Transport Module of the new DELFIC fallout prediction system has undergone additional development since publication of its description in DASA 2669. This supplement to DASA 2669 describes these developments and presents ammendments and corrections to the code and its documentation. Complete FORTRAN statement listings of subroutines that have been substantially changed are included.

TABLE OF CONTENTS

1. INTRODUCTION

Since publication of DASA 2669*, development and application of the DELFIC Diffusive Transport Module (DTM) has continued. A few revisions to the model have been made, several of them of major importance. In addition, some hidden "bugs" have been uncovered and corrected. In this supplement to DASA 2669 we describe the important model revisions, and amend the documentation. We also correct errors in the documentation, and provide FORTRAN statement listings of subroutines that have been changed substantially.

* H. G. Norment and E. J. Tichovolsky, "A New Fallout Transport Code for the DELFIC System: The Diffusive Transport Module," ARCON Corporation Report R71-1W, DASA 2669 (1 March 1971), AD 727 613.

2. MODEL REVISIONS

2.1 Initial Parcel Description

Fallout parcels are taken to be distributed in the horizontal about their centers of mass by a Gaussian density function. The initial Gaussian standard deviation was set equal to the parcel radius that was received via tape IPARIN from the Cloud Rise-Transport Interface Module. This has been changed so that the initial standard deviation is one-half the input parcel radius. (Compare subroutine SPRVS card 81 of DASA 2669 with SPRVS card 94 of this supplement.)

2.2 Simple Advection-Plus-Settling

In some cases, it is desirable to transport fallout in a simple advection-plus-settling mode; that is, without accounting for diffusion in the vertical. In this mode, integration of Eq. (16) is bypassed, and the parcel trajectory is computed via Eq. (32). As actually employed in the original DTM, Eq. (32) was modified to the form

$$\vec{r}_c = \vec{r}_i + \frac{1}{\langle f\rangle - \langle w\rangle} \sum_{z_i}^{z_g} \vec{U}(x,y,z,t)\Delta z \ ,$$

where the average settling speed, $\langle f\rangle$, was taken to be

$$\langle f\rangle = \frac{f(z_i) + f(z_g)}{2} \ ,$$

and $\langle w\rangle$ was an average vertical air velocity. For cases where $z_i - z_g$ is large, there can be significant differences in the particle settling speeds in the upper layers compared with those in the lower layers. When, in addition, there is large wind shear in the vertical,

Preceding page blank

it becomes necessary to use settling speeds that are computed individually for each wind layer instead of an average settling speed. The code has been changed so that this is done.

Major changes have been required in several programs. The most extensive changes are in subroutines SPRVS and TRANP. A new array, CAVS(KBHF), is created to store a table of particle settling speeds; it contains an entry for each wind layer.

Before a parcel is transported via this mode, a test is made to determine if the parcel can impact in the time allowed for transport. This test is simplified by using the knowledge that all parcels comprised of particles of a particular size class are processed sequentially in one group. Thus, as each new particle size class is encountered, the altitude above which these particles cannot impact is computed. Then, any parcel in the group whose base is above this altitude is bypassed. To perform the altitude limit calculation, an average vertical wind velocity is needed for each wind layer. To accommodate this, the array WAVG(LTIMF) was changed to the array WAVG(KBHF,LTIMF).

3. DOCUMENTATION REVISIONS

Revisions and corrections are intermixed and listed in order of their encounter in DASA 2669.

Page 27, Eq. (23):

$$t_1 = 4c_3 \left(\frac{\sigma_{t_i}^2}{\varepsilon} \right)^{1/3} \tag{23}$$

Page 27, line 16:

In our application, σ_{t_i} is taken to be one-half of the radius of a cloud wafer as recorded

Page 44, line 3:

where the summation is over the N data with the largest f_i, and the weighting factors,

Page 44, lines 6 and 7:

The parameters α, β, and N are specified by the user and the x_i and y_i are relative to the n-th lattice cell center. The calculations of the f_i are performed so that whenever a factor in Eq. (39) is found to

Page 44, lines 9 and 10:

than the total number of observations, M, only the N observations with the largest f_i relative to the n-th lattice cell center are considered in the calculations.

Page 45, lines 8-12:

list, two tables of settling rates for this particle are computed. One table contains an entry for each altitude increment used in the numerical integration of Eq. (16).

The other table contains an entry for each of the larger altitude increments that are used to resolve the wind and turbulence fields. These same tables are used through transport for this parcel and the remainder of the parcels in the first group. When the first parcel of the second group is selected, new tables are computed, and so on.

Page 49, Eq. (45):

$$\frac{G_j^{n+1} - G_j^n}{\Delta t} = \frac{1}{2(\Delta z)^2} \Big[(K_{j+1} + K_j)G_{j+1}^n - (K_{j+1} + 2K_j + K_{j-1})G_j^n + (K_j + K_{j-1})G_{j-1}^n\Big] + \frac{1}{\Delta z}\Big[(f_{j+1} - w_{j+1})G_{j+1}^n - (f_j - w_j)G_j^n\Big] \quad (45)$$

Page 53, following Eq. (65):

where in the code K_{J+1} is taken to be equal to K_J.

Page 66, paragraphs 1 and 2:

Beginning at its input location and time, the parcel base or top is transported via local winds in the cell of its residence. At the same time it settles at a speed that is computed for the altitude at the center of the wind cell of its residence. When it passes through a wind cell boundary or a time boundary, the wind and settling speed are changed to those of the new cell or update. This continues until ground impaction occurs or until an extreme wind field or time boundary is encountered. The calculation requires one step for each cell through which the base or top passes.

When an extreme boundary is encountered by a top or base, the location and time of the encounter is recorded. These values are used in the definition of the deposit increment as described on page 70. The altitude of a deposit increment is always recorded as the arithmetic average of the impact

altitudes of its top and base. Thus, the recorded altitude of a deposit increment that has reached an extreme boundary can be well above the deposition plane.

Page 71, Figure 9:

The quantity labeled $\sigma(\underline{i})$ that lies to the left of the deposit increment ellipse should be replaced by $\sigma(\underline{i})_d$.

Page 88, lines 10 and 11:

DFZ. An area-weighted average vertical wind, WAVG(KBH, LTIM), is derived from array WFZ for each altitude layer and update. Likewise, a volume-weighted

Page 88, equation for DKAV:

$$\text{DKAV} = \sum_{\text{KBH}=1}^{\text{KBHX}-1} \frac{\text{DFZ(KBH,NDATA,LTIM)*(ZBH(KBH+1) - ZBH(KBH))}}{\text{ZBH(KBHX) - ZBH(1)}}$$

Page 88, lines 27 and 28:

out on ISOUT. In a parallel operation the quantities WAVG(KBH,LTIM), which are area-weighted vertical wind velocities for each wind layer and update, are computed and printed out on

Page 96, line 25:

in the horizontal RWFR(J)/2.0;

page 97, line 12:

ZPAR(J), PSAM(J), RWFR(J)/2.0, DWFR(J), ZLWF(J), and VWFR(J) are

Page 98, line 2:

for further details.) Also computed are the settling speeds,

CAVS(KBH), for each altitude, and the altitude, ZLIM, above which deposition is impossible, via gravity settling in the specified wind field, for particles in the particular size class being considered.

Page 100, last line:

CAV - CAV - WAVGK

Page 101, first line:

WAVGK and DAVG(1) are the average vertical components of wind

Page 101, line 15:

ground by advection at fall rates CAVS(KBHF) via a call to subroutine ADVEC.

Page 101, line 23:

limit: i.e., whenever ZLOW $\geq$ ZLIM. The comment

Page 110, lines 16-19:

also obtained via the COMMON area QPARM. The particle settling speeds, CAVS(KBHF), and area-weighted vertical air velocities, WAVG(KBHF,LTIMF), for each wind layer, are obtained

Page 110, line 27:

parcel base is advected, while settling at speeds CAVS(KBH) - WAVG(KBH,LTIM), from position (XP,YP,ZP =

Page 110, line 29:

TOL. The standard deviations of the

Page 110, line 33:

parcel top is advected, while settling at speeds CAVS(KBH) - WAVG(KBH,LTIM), from position (XP,YP,

Page 111, line 2:

at time TOU. The standard

Page 111, second paragraph:

After both base and top of a parcel have been transported, the arithmetic average of the two impact times and impact altitudes are recorded for the deposit increment (as shown on the next page). In the event that the base or top encounters an extreme wind field or time boundary during transport, subroutine TRANP returns the coordinates of the encounter point. Therefore, the altitude recorded for a deposit increment can be well above the deposition plane.

Page 111, third paragraph, lines 4 and 5:

Is considered superfluous whenever ZP differs from ZDEP by less than 0.1. Instead, XOL, YOL, ZOL, TOL, SIGXL, SIGYL,

Page 122, line 28:

When vertical diffusive transport is employed, the computation of horizontal parcel advection is based on the

Page 123, insert between the second and third paragraphs:

For transport via simple advection plus settling, TRANP is called by subroutine ADVEC with KRIP=0 (otherwise KRIP=1). The parcel top or bottom is transported stepwise through the vertical layers between ZP and ZDEP via the layerwise mode (see below). In each layer, with index KBH, the vertical velocity is

WBAR = WFZ(KBH, NDATO, .TIM) - CAVS(KBH)

and KAY = - .1. TSEG is computed as shown by the equation in the preceding paragraph.

Page 123, third paragraph, line 3:

is traversed at a time. This mode is mandatory for transport via simple advection plus settling, or when a parcel trajectory

Page 125, line 6:

The rapid computation mode is employed for vertical diffusive transport when

Page 145, insert at the end of the Input Data Card 4 discussion:

To preempt vertical diffusive transport, set KX=1 and set ZMAX arbitrarily large. This causes all parcels to be transported via the simple-advection-plus-settling mode (Eq. (32)). Of course, horizontal diffusive growth of parcels is accounted for in any case.

Page 146, line 14:

CSKIP - 0.1

Page 152, line 8:

HITIME. If the vertical diffusive mode of transport is used, the KBHX'th base should be above or at the top of the transport space as this top is specified by ZMAX.

Page 156, Table 4, Record Number 12, line 5 under Content:

size class central diameter (μm), mass of fallout (kg)

Page 154, to the end of paragraph 1 add:

However, the turbulent energy dissipation rates can be input only for the horizontal directions; for the vertical direction Fickian diffusivities always are input, regardless of which type of data are input for the horizontal.

4. CODE REVISIONS

4.1 Single Card Changes

Addition of the arrays CAVS(KBHF) and WAVG(KBHF,LTIMF) requires revisions in DIMENSION statements and subroutine argument lists. However, complete FORTRAN statement listings are given in this supplement for all subroutines that require these revisions.

Subroutine BOUN, card 27:

```
CALL NEST (NET, NETSU, XO, YO, NDATO, XL, XR, YL, YU, ICF,
JCF, NCF)
```

Subroutine DUMPER:

Place card 33 in its proper position.

Subroutine NEST, insert between cards 20 and 21:

```
DIMENSION NET(ICF, JCF), NETSU(NCF)
```

4.2 FORTRAN Statement Listings

Complete FORTRAN statement listings are given for the following subroutines. These subroutines are operational on the UNIVAC 1108.

The machine used to prepare these listings prints a # symbol to represent a 4-8 punch; this symbol should be an apostrophe ('). In FORMAT and DATA statements, the apostrophe is used to define Hollerith character fields.

```
C     SEPTEMBER 1971
C     C31M IS THE MAIN PROGRAM WHICH DIRECTS THE DIFFUSIVE TRANSPORT
C     MODULE OPERATIONS.THE OBJECT-TIME DIMENSIONS ARE SET IN C31M.
C     THESE DIMENSIONS AND THEIR RESPECTIVE ARRAYS ARE
C     KKF   - AA,BB,CC,DENOM,E,F,Q,W,DIFF,ZHT
C     LTIMF - TIMUP,DAVG,WAVG
C     KBHF  - ZBH,ZCH
C     NDATF - KTOPO
C     KBHF, NDATF, LTIMF - DFZ,DXSUM,DYSUM,USUM,VSUM,WFZ,RSUM
C     ICF, JCF - NET
C     NCF   - NETSU
C     MARF  - MARY
C     NATF  - ALT,ATEMP,RHO
C     DATA LITERALS MUST BE INSERTED IN THE DIMENSION STATEMENTS AND IN
C     THE RIGHT HAND SIDES OF THE ARITHMETIC STATEMENTS IN WHICH THE
C     ABOVE VARIABLE NAMES APPEAR.
C     ************************** GLOSSARY ******************************
C     AA(K) - EQUALS S2*(DIFF(K+1)+DIFF(K))+S1*F(K+1)
C     ALT   - ALTITUDES FOR ATMOS. DENSITY AND VISCOSITY TABLE
C     ATEMP - DYNAMIC VISCOSITY OF AIR DATA VECTOR FOR ATMOS. TABLE
C     BB(K) - EQUALS S2*(DIFF(K+1)+2.*DIFF(K)+DIFF(K-1))+S1*F(K)
C     CAV   - AVG. FALLRATE USED IN COMPUTING PDEST. IT APPLIES MID WAY
C             FROM PARCEL TOP TO ZMIN.
C     CAVS  - PARTICLE FALL RATE FOR EACH ATMOS. STRATUM
C     CC(K) - EQUALS S1*(DIFF(K)+DIFF(K-1))
C     CROSS - CROSSWIND CROSSING TRAJECTORIES CORRECTION TO TURB.
C     CSKIP - TOTAL FRACTIONAL PARCEL DEPOSITION THRESHOLD
C     DAVG  - AVG. ATMOS. VERT. TURB. PER UPDATE DATA VECTOR
C     DENOM(K) - EQUALS 1.-THETA*(BB(K)-CC(K)*E(K-1))
C     DEP   - DEPOSITED FRACTIONAL MASS INCREMENT
C     DIFF(K) - VERT. DIFFUSIVITY AT K-TH SMALL ALTITUDE INCREMENT
C     DFKXS1- VERTICAL DFFUSIVITY AT ALTITUDE INCREMENT KX-1
C     DFZ   - TURBULENCE Z COMPONENT 3-DIM. DATA ARRAY
C     DINCR - RATE OF CHANGE OF FRACTIONAL MASS DEPOS. RATE THRESH.
C     DOPEN - MASS DEPOSITION RATE THRESHOLD
C     DOWN  - DOWNWIND CROSSING TRAJECTORIES CORRECTION TO TURB.
C     DT    - SMALL ITERATION TIME STEP FOR VERT. DIFF. DIFF. EQ.
C     DWAF  - PARCEL VERT. THICKNESS BEFORE ADVECTION
C     DXSUM - TURBULENCE X COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     DYSUM - TURBULENCE Y COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     DZ    - SMALL ALTITUDE INCREMENT FOR VERT. DIFF. DIFF. EQ.
C     DZMIN - MINIMUM VALUE OF DZ
C     E(K)  - EQUALS THETA*AA(K)/DENOM(K)
C     EDDY  - RATIO OF LAGRANGIAN TURBULENCE TIME SCALE TO EULERIAN
C             TURBULENCE LENGTH SCALE
C     EFFLUX  UPPER EFFLUX FRACTIONAL MASS
C     F(K)  - IN SUB. DIFFF, WORKING SPACE IMPLICIT METHOD DATA VECTOR.
C             IN SUB. AMBNT, WORKING SPACE FOR VERTICAL VELOCITIES
C     FAV   - MID-ATMOS. AVG. FALLRATE. USED IN CROSSING TRAJECTORIES
C             CORRECTIONS AND IN TRUNCATION ERROR ESTIMATION .
C     FMAB  - CUMULATIVE FRACTIONAL MASS AIRBORNE
C     FMBEL - MIN. PARCEL FRACTIONAL MASS ALOFT TO BE TRANSPORTED
C     ICF   - MAX. FORMAL DIM. CORRESPONDING TO ICX
C     ICX   - OBJECT-TIME FIRST MAX. DIM. OF ARRAY NET. NUMBER OF NET
C             MESHES IN EAST-WEST ROW.
C     IMARIN- LOGICAL UNIT NUMBER OF CR-TRANS. INTER. MOD. OUTPUT TAPE
C     IPOUT - LOGICAL UNIT NUMBER OF DIFF. TRANS. MOD. OUTPUT TAPE
C     ISIN  - LOGICAL UNIT NUMBER OF SYSTEM INPUT TAPE
```

```
C     ISOUT - LOGICAL UNIT NUMBER OF SYSTEM OUTPUT TAPE
C     JCF   - MAX. FORMAL DIM. CORRESPONDING TO JCX
C     JCX   - OBJECT-TIME SECOND MAX. DIM. OF ARRAY NET. NUMBER OF NET
C             MESHES IN SOUTH-NORTH ROW.
C     KBH   - ATMOS. VERT.  SPACE INDEX FOR ARRAYS USUM,VSUM,WFZ,DXSUM,
C             DYSUM, DFZ, RSUM, ZBH, ZCH
C     KBHF  - MAX. FORMAL DIM. CORRESPONDING TO KBHX
C     KBHX  - OBJECT-TIME MAX. VALUE OF KBH
C     KKF   - MAX. FORMAL DIM. CORRESPONDING TO KKX
C     KKM   - ALWAYS EQUALS 2. CORRESPONDS TO K=0 ALTITUDE INCREMENT
C     KKMA1 - EQUALS KKM+1
C     KKX   - EQUALS KX+KKM
C     KKXS1 - EQUALS KKX-1
C     KRIP   - CONTROL VARIABLE
C                0 FOR ADVECTIVE TRANSPORT
C                1 FOR DIFFUSIVE TRANSPORT
C     KTOPO - NET MESH AND SUB-MESH TOPOGRAPHY TABLE DATA VECTOR
C     KX    - MAX. NUMBER OF DZ ALTITUDE INCREMENTS
C     KXMIN - MIN. NUMBER OF DZ ALTITUDE INCREMENTS
C     LSTEP - NUMBER OF IMPLICIT METHOD ITERATIONS. TDELT=LSTEP*DT.
C             DYSUM, DFZ, RSUM, TIMUP, DAVG, HAVG
C     LTIM  - ATMOS. UPDATE TIME  INDEX FOR ARRAYS USUM,VSUM,WFZ,DXSUM,
C     LTIMX - OBJECT-TIME MAX. VALUE OF LTIM
C     LTIMF - MAX. FORMAL DIM. CORRESPONDING TO LTIMX
C     MARF  - MAX. FORMAL DIM. CORRESPONDING TO MARX
C     MARX  - OBJECT-TIME MAX. DIM. OF ARRAY MARY
C     MARY  - HORIZ. ATMOS. SPACE RESOLUTION NET MESH AND SUB-MESH
C             CONTROL FLAGS DATA VECTOR
C
C     MC    - CONTROL INTEGER DATA VECTOR
C
C             MC(1) LESS THAN OR EQUAL TO ZERO, SUPPRESSES LISTING OF
C             EXPANDED WIND AND TURB. DATA
C
C             MC(1) GREATER THAN OR EQUAL TO ONE, CAUSES LISTING OF
C             WIND AND TURB. DATA BEFORE SUMMATION
C
C             MC(1) EQUALS TWO, CAUSES LISTING OF WIND AND TURB. DATA
C             AFTER SUMMATION
C
C             MC(2) EQUALS ONE, SUPPRESSES LISTING OF ATMOS. VISC. AND
C             DENS. TABLES
C
C             MC(3) EQUALS ZERO, SUPPRESSES LISTING OF DEPOSIT
C             INCREMENTS ON TAPE ISOUT
C
C             MC(4) EQUALS ONE, CAUSES PRINTOUT OF TRANSPORT
C             INTERMEDIATE RESULTS ON TAPE ISOUT. WARNING. PRINTOUT
C             IS EXTRAORDINARILY VOLUMINOUS. FOR DEBUGGING ONLY.
C
C             MC(7) EQUALS ONE, SUPPRESSES LISTING OF RAW WIND AND TURB.
C             INPUT DATA
C
C             MC(10) EQUALS ONE, CAUSES TURB. DATA TO BE TREATED AS
C             KOLMOGOROFF-BATCHELOR ENERGY DISSIPATION RATES
C
C             MC(10) NOT EQUAL TO ONE, CAUSES TURB. DATA TO BE TREATED
C             AS FICKIAN DIFFUSIVITIES
```

```
C
C           MC(18) EQUALS ONE, SUPPRESSES READING FROM TAPE IPARIN
C           AND WRITING ONTO TAPE IPOUT
C
C     MINT   - MIN. NUMBER OF DT SMALL TIME STEPS PER DEPOSIT INCREMENT
C              TIME INTERVAL
C     NAT    - NUMBER OF ALTITUDE STRATA IN ATMOS. DENS. AND VISC. TABLE
C     NATF   - MAX. FORMAL DIM. CORRESPONDING TO NATF
C     NBLK   - RECORD BLOCK SIZE FOR DEPOSIT INCREMENT RECORDS
C     NCF    - MAX. FORMAL DIM. CORRESPONDING TO NCX
C     NCX    - OBJECT-TIME MAX. DIM. OF ARRAY NETSU
C     NDATA  - ATMOS. HORIZ. SPACE INDEX FOR ARRAYS USUM,VSUM,WFZ,DXSUM,
C              DYSUM, DFZ, RSUM, KTOPO
C     NDATC  - HORIZONTAL INDEX OF LATTICE CELL CONTAINING POINT (XC,YC)
C     NDATO  - HORIZONTAL INDEX OF LATTICE CELL CONTAINING POINT (XO,YO)
C     NDATP  - HORIZONTAL INDEX OF LATTICE CELL CONTAINING POINT (XP,YP)
C     NDATX  - OBJECT-TIME MAX. VALUE OF NDATA
C     NDATF  - MAX. FORMAL DIM. CORRESPONDING TO NDATX
C     NDELT  - NOMINAL NUMBER OF DEPOSIT INCREMENTS PER FALLOUT PARCEL
C     NET    - HORIZONTAL SPACE CONTROL NET MESH 2-DIM. ARRAY
C     NETSU  - HORIZONTAL SPACE CONTROL NET SUB-MESH DATA VECTOR
C     NSEQC  - STORAGE SEQUENCE ORDINAL OF FIRST PARCEL TO BE TRANSPORTED
C     PHI    - EQUALS 1-THETA
C     Q(K)   - CONCENTRATION IN K-TH ALTITUDE INCREMENT
C     RHO    - ATMOS. DENSITY DATA VECTOR FOR ATMOS. TABLE
C     RO     - WIND HEADING ORIENTATION ANGLE AFTER ADVECTION
C     ROPART - FALLOUT PARTICLE DENSITY
C     RSUM   - WIND HEADING ORIENTATION ANGLE (WEIGHTED SUM) 3-DIM. ARRAY
C     RWAF   - PARCEL RADILS IN PARCEL CENTRAL PLANE BEFORE ADVECTION
C     SIGXO  - PARCEL MASS HOR. STAND. DEV. DOWNWIND AFTER ADVECTION
C     SIGYO  - PARCEL MASS HOR. STAND. DEV. CROSSWIND AFTER ADVECTION
C     S1     - EQUALS DT/DZ
C     S2     - EQUALS DT/(2.*(DZ)**2))
C     TDELT  - CURRENT DEPOSIT INCREMENT TIME INTERVAL
C     TDEP   - ADVECTIVE TRANSPORT TIME INTERVAL
C     THETA  - IMPLICIT FINITE DIFFERENCE PARAMETER
C     THETD  - DOUBLE PRECISION WORD CORRESPONDING TO THETA
C     TIME   - TIME AT ONSET OF CURRENT DEPOSIT INCREMENT TIME INTERVAL
C     TIMEX  - SIMULATED TRANSPORT TIME LIMIT
C     TIMUP  - ATMOSPHERE UPDATE TIMETABLE DATA VECTOR
C     TO     - TIME AFTER PARCEL ADVECTION
C     TP     - TIME BEFORE PARCEL ADVECTION
C     TPAUS  - TIME AT END OF CURRENT DEPOSIT INCREMENT TIME INTERVAL
C     USUM   - WIND X COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     VETA   - VERTICAL DIFFUSION ABSORPTION COEFFICIENT
C     VSUM   - WIND Y COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     W(K)   - SETTLING RATE AT K-TH SMALL ALTITUDE INCREMENT
C     WAVG   - AVG. ATMOS. VERT. WIND  PER UPDATE PER STRATUM
C     WAVGK  - WAVG AVERAGED OVER THE STRATA FOR THE FIRST UPDATE
C     WFZ    - WIND Z COMPONENT 3-DIM. DATA ARRAY
C     WINT   - NET CONTROL MESH DIMENSION
C     XLLC   - X COORDINATE OF SOUTH-WEST CORNER OF ATMOS. SPACE
C     XO     - PARCEL CENTER X COORDINATE AFTER ADVECTION
C     XP     - PARCEL CENTER X COORDINATE BEFORE ADVECTION
C     YLLC   - X YOORDINATE OF SOUTH-WEST CORNER OF ATMOS. SPACE
C     YO     - PARCEL CENTER Y COORDINATE AFTER ADVECTION
C     YP     - PARCEL CENTER Y COORDINATE BEFORE ADVECTION
C     ZBH    - ATMOSPHERE STRATA BASE-ALTITUDE DATA VECTOR
```

```
C      ZCH   - ATMOSPHERE STRATA MID-ALTITUDE DATA VECTOR
C      ZDEP  - ADVECTIVE TRANSPORT TERMINAL ALTITUDE
C      ZHT(K) - K-TH ALTITUDE INCREMENT ABOVE ZMIN LEVEL
C      ZLOW  - PARCEL BASE ALTITUDE BEFORE ADVECTION
C      ZMAX  - ATMOSPHERE TOP ALTITUDE (FOR VERT. DIFF. DIFF. EQ.)
C      ZMIN  - GROUND LEVEL OR DEPOSITION PLANE ALTITUDE
C      ZO    - PARCEL CENTER Z COORDINATE AFTER ADVECTION
C      ZP    - PARCEL CENTER Z COORDINATE BEFORE ADVECTION, ECXEPT AS
C                REDEFINED IN SUB. ADVEC
C      ZUPP  - PARCEL TOP ALTITUDE BEFORE ADVECTION. ZLOW+DWAF
       DOUBLE PRECISION AA(204),BB(204),CC(204),DENOM(204),E(204),F(204)
       DOUBLE PRECISION Q(204)
       DIMENSION ZBH(     27),ZCH(     27),TIMUP(      6),MARY(       1)
       DIMENSION KTOPO(      1),NETSU(      1),NET(      1,       1)
       DIMENSION   DFZ(     27,       1,       6),  WFZ(     27,       1,       6)
       DIMENSION  USUM(     27,       1,       6), VSUM(     27,       1,       6)
       DIMENSION DXSUM(     27,       1,       6),DYSUM(     27,       1,       6)
       DIMENSION  RSUM(     27,       1,       6),DAVG(      6)
       DIMENSION  WAVG(     27,       6)
       DIMENSION ALT(    260),RHO(    260),ATEMP(    260)
       DIMENSION CAVS(    27),W(    204),DIFF(    204),ZHT(    204)
       COMMON /QPARM/ IPOUT ,IPARIN,NBLK  ,NAT   ,NDELT ,KX    ,KKM
      1,NSEQO ,ICX   ,JCX   ,NCX   ,KBHX  ,NDATX ,LTIMX ,ISIN  ,ISOUT
      2,EDDY  ,FMREL ,LSTEF ,MC(13),WINT  ,XLLC  ,YLLC  ,THETA ,ZMIN
      3,CSKIP ,MINT  ,ZMAX  ,TIMEX ,DT    ,DZ    ,XP    ,YP    ,ZP
      4,DINCR ,DOWN  ,TP    ,ZLOW  ,DWAF  ,PWAF  ,ROPART,ZUPP  ,VETA
      5,DOPEN ,CROSS ,TIME  ,KKMA1 ,KKX   ,KKXS1 ,KXMIN ,NDATP
       ICF=1
       ISIN=5
       ISOUT=6
       IPARIN=9
       IPOUT=10
       JCF=1
       KBHF=27
       KKF=204
       LTIMF=6
       MARF=1
       NATF=260
       NCF=1
       NDATF=1
       DO 1 N=1,NCF
     1 NETSU(N)=0
       DO 2 J=1,JCF
       DO 2 I=1,ICF
     2 NET(I,J)=0
       DO 3 M=1,MARF
     3 MARY(M)=0
       DO 4 K=1,KBHF
       ZBH(K)=0.
     4 ZCH(K)=0.
       DO 104 K=1,KBHF
       DO 104 L=1,LTIMF
   104 WAVG(K,L)=0.0
       DO 5 N=1,NDATF
     5 KTOPO(N)=0
       DO 6 L=1,LTIMF
       TIMUP(L)=0.
       DAVG(L)=0.
```

```
      DO 6 N=1,NDATF
      DO 6 K=1,KBHF
      DFZ(K,N,L)=0.
      WFZ(K,N,L)=0.
      USUM(K,N,L)=0.
      VSUM(K,N,L)=0.
      DXSUM(K,N,L)=0.
      DYSUM(K,N,L)=0.
    6 RSUM(K,N,L)=0.
COMMENCE READING DATA INPUTS FROM TAPES ISIN AND IPARIN
COMMENCE WRITING DATA OUTPUT HEADERS ONTO TAPES ISOUT AND IPOUT
      CALL        LINK(ALT,RHO,ATEMP,NATF)
CONSTRUCT THE HORIZONTAL SPACE CONTROL NET
      CALL        GETUP(NET,NETSU,KTOPC,MARY,MARF,ICF,JCF,NCF,NDATF)
CONSTRUCT THE ATMOSPHERIC STRATA AND UPDATE DATA VECTORS
      CALL        HITIMF(ZCH,ZBH,TIMUP,KBHF,LTIMF)
CONTSRUCT AND FILL IN THE ATMOSPHERIC LATTICE AND UPDATE STRUCTURE
      CALL        ADMIN(NET,NETSU,ZBH,ZCH,TIMUP,USUM,VSUM,DXSUM,DYSUM,
     1RSUM,DFZ,WFZ,DAVG,WAVG,ICF,JCF,NCF,KBHF,NDATF,LTIMF)
CIRCUMVENT ALL TAPE HANDLING IF MC(18) EQUALS 1
      IF(MC(18).EQ.1) GO TO 7
CALCULATE THE DIFFUSIVE TRANSPORT OF PARCELS ACCEPTED FROM TAPE IPARIN
COPY OUT RESULTS ONTO TAPE IPOUT
      CALL        SPRVS(NET,NETSU,ZBH,ZCH,TIMUP,USUM,VSUM,DXSUM,DYSUM,
     1RSUM,DFZ,WFZ,DAVG,WAVG,ALT,RHO,ATEMP,AA,BB,CC,DENOM,DIFF,E,F,Q,W,
     2ZHT,ICF,JCF,NCF,KBHF,NDATF,LTIMF,KKF,NATF,CAVS)
    7 CALL EXIT
      STOP
      END
```

```
      SUBROUTINE ADMIN(NET,NETSU,ZBH,ZCH,TIMUP,USUM,VSUM,DXSUM,DYSUM,
     1RSUM,DFZ,WFZ,DAVG,WAVG,ICF,JCF,NCF,KBHF,NDATF,LTIMF)
C     SEPTEMBER 1971
C
C     SUBROUTINE ADMIN CONSTRUCTS WIND DATA ARRAYS
C                USUM, VSUM, WFZ, WAVG, RSUM
C     AND TURBULENCE DATA ARRAYS
C                DXSUM, DYSUM, DFZ, DAVG.
C     IN ADMIN ONLY LTIM AND SPEC ARE READ FROM TAPE ISIN.
C     SPEC  - DATA SPECIES IDENTIFICATION WORD ≠WIND≠ OR ≠SPEC≠
C     LTIM  - UPDATE ORDINAL OF DATA SET. FIRST ATMOS. SET  HAS LTIM=1.
C     SUB. MKDAT IS CALLED TO PERFORM DATA EXTRAPOLATIONS.
C     AREA  - AREA OF HORIZ. SPACE NET
C     AREAN - AREA OF N-TH NET MESH OR SUB-MESH
      COMMON /OPARM/ IPOUT ,IPARIN,NBLK  ,NAT   ,NDELT ,KX    ,KKM
     1,NSEQ ,ICX   ,JCX   ,NCX   ,KBHX  ,NDATX ,LTIMX ,ISIN  ,ISOUT
     2,EDDY  ,FMBEL ,LSTEP ,MC(18),WINT  ,XLLC  ,YLLC  ,THETA ,ZMIN
     3,CSKIP ,MINT  ,ZMAX  ,TIMEX ,DT    ,DZ    ,XP    ,YP    ,ZP
     4,DINCR ,DOWN  ,TP    ,ZLOW  ,DWAF  ,RWAF  ,ROPART,ZUPP  ,VETA
     5,DOPEN ,CROSS ,TIME  ,KKMA1 ,KKX   ,KKXS1 ,KXMIN ,NDATP
      DIMENSION RSUM(KBHF,NDATF,LTIMF),DAVG(LTIMF),WAVG(KBHF,LTIMF)
      DIMENSION NET(ICF,JCF),NETSU(NCF),ZCH(KBHF),TIMUP(LTIMF),ZBH(KBHF)
      DIMENSION USUM(KBHF,NDATF,LTIMF),VSUM(KBHF,NDATF,LTIMF)
      DIMENSION DXSUM(KBHF,NDATF,LTIMF),DYSUM(KBHF,NDATF,LTIMF)
      DIMENSION DFZ(KBHF,NDATF,LTIMF),WFZ(KBHF,NDATF,LTIMF)
      DIMENSION LW(10),LD(10)
      INTEGER WIND,DFSN,DONE,SPEC
      DATA PROGRM/≠ADMIN ≠/
      DATA WIND/≠WIND≠/
      DATA DFSN/≠DIFF≠/
      DATA DONE/≠NO M≠/
    1 FORMAT(≠0≠36X,≠UPDATE TIME INDEX≠I5,≠. WIND GRID CELL INDEX≠I5/)
    2 FORMAT(≠  WIND≠2(6X,≠HORIZONTAL≠),6X,≠ VERTICAL ≠6X,≠CROSSWIND ≠6X
     1,≠ DOWNWIND ≠6X,≠ VERTICAL ≠6X,≠HORIZONTAL≠)
    3 FORMAT(≠ LAYER≠6X,≠E.-W. WIND≠6X,≠N.-S. WIND≠6X,≠   WIND   ≠3(6X,≠
     2DIFFUSION ≠),6X,≠ ROTATION≠)
    4 FORMAT(≠ INDEX≠3(6X,≠ VELOCITY ≠),3(6X,≠ CONSTANT ≠),6X,≠  ANGLE≠)
    5 FORMAT(≠ ≠I5,7E16.4)
    6 FORMAT(/25X,≠WEIGHTED SUMS OVER ABOVE COLUMN ENTRIES≠/)
    7 FORMAT(≠ ≠I5,2E16.4,16X,2E16.4,16X,E16.4)
    8 FORMAT(≠ ≠22X,≠UPDATE≠I4,≠* CELL≠I4,                         ≠*
     8AVG. VERT. DIFF. =≠E12.4)
    9 FORMAT(36X,I2,8X,3A4)
   10 FORMAT(/25X,≠ATMOSPHERE UPDATE≠I4,≠ FOR TIMES LATER THAN ≠E12.4,≠
     1SECONDS≠)
   11 FORMAT(≠ ≠25X,≠* * * * * * * * * * * WINDFIELD EXTRAPOLATION * * * *
     1 * * * * * *≠/)
   12 FORMAT(≠ ≠25X,≠* * * * * * * * * * DIFFUSIVITY EXTRAPOLATION * * * *
     2 * * * * * *≠/)
   13 FORMAT(/25X,≠UPDATE≠I4,≠ OF THE WINDFIELD IS MISSING≠)
   14 FORMAT(/25X,≠UPDATE≠I4,≠ OF THE DIFFUSIVITY IS MISSING≠)
   15 FORMAT(≠ OVER ENTIRE HORIZONTAL GRID FOR UPDATE≠I4,
     6           ≠* AVG. VERT. DIFF. =≠E12.4)
   16 FORMAT(≠  AVG. VERT. VEL. FOR EACH STRATUM IS - ≠)
   17 FORMAT(2X,I5,E16.4)
      AREA=ICX*JCX*(WINT**2)
      DO 999 L=1,LTIMX
      LW(L)=L
```

```
  999 LD(L)=L
COPY IN LTIM AND SPEC FROM TAPE ISIN AND CALL SUB. MKDAT
 1000 READ(ISIN,9) LTIM,SPEC
      IF(SPEC.EQ.DCNE) GO TO 1301
      IF((LTIM.LT.1).OR.(LTIM.GT.LTIMX)) CALL ERROR(PROGRM,-1000,ISOUT)
      WRITE(ISOUT,10) LTIM,TIMUP(LTIM)
      IF(SPEC.EQ.WIND) GO TO 1101
      IF(SPEC.EQ.DFSN) GO TO 1201
      CALL ERROR(PROGRM,-1101,ISOUT)
 1101 WRITE(ISOUT,11)
      DO 1102 L=1,LTIMX
 1102 IF(LTIM.EQ.LW(L)) GO TO 1103
      CALL ERROR(PROGRM,-1102,ISOUT)
 1103 LW(L)=-1
      CALL      MKDAT(ZCH,NET,NETSU,LTIM, USUM, VSUM,WFZ,ICF,JCF,NCF,
     1KBHF,NDATF,LTIMF)
      GO TO 1000
 1201 WRITE(ISOUT,12)
      DO 1202 L=1,LTIMX
 1202 IF(LTIM.EQ.LD(L)) GO TO 1203
      CALL ERROR(PROGRM,-1202,ISOUT)
 1203 LD(L)=-1
      CALL      MKDAT(ZCH,NET,NETSU,LTIM,DXSUM,DYSUM,DFZ,ICF,JCF,NCF,
     1KBHF,NDATF,LTIMF)
      GO TO 1000
CHECK IF ANY WIND DATA SETS ARE MISSING
 1301 DO 1303 L=1,LTIMX
      IF(LW(L).EQ.-1) GO TO 1302
      WRITE(ISOUT,13) LTIM
      CALL ERROR(PROGRM,-1302,ISOUT)
CHECK IF ANY TURBULENCE DATA SETS ARE MISSING
 1302 IF(LD(L).EQ.-1) GO TO 1303
      WRITE(ISOUT,14) LTIM
      CALL ERROR(PROGRM,-1303,ISOUT)
 1303 CONTINUE
CALCULATE THE WEIGHTED SUMS OVER ATMOS. STRATA AND REWRITE ARRAYS
C     USUM, VSUM, RSUM, DXSYM, DYSUM. ALSO COMPUTE DAVG AND WAVG.
      ZSPAN=ZBH(KBHX)-ZBH(1)
      DO 922 L=1,LTIMX
      DO 1304 LK=1,KBHX
 1304 WAVG(LK,L)=0.0
      DAVG(L)=0.
      DO 921 N=1,NDATX
      DKAV=0.
      IF(MC(1).LT.1) GO TO 915
      WRITE(ISOUT,1) L,N
      WRITE(ISOUT,2)
      WRITE(ISOUT,3)
      WRITE(ISOUT,4)
 915  DO 920 K=1,KBHX
      UKNL=USUM(K,N,L)
      VKNL=VSUM(K,N,L)
      IF(ABS(UKNL)-1.0E-30) 9151,9154,9154
 9151 IF(ABS(VKNL)-1.0E-30) 9152,9153,9153
 9152 RKNL=0.
      GO TO 9155
 9153 RKNL=1.57079633
      GO TO 9155
```

```
9154 RKNL=ATAN(VKNL/UKNL)
9155 DXKNL=DXSUM(K,N,L)
     DYKNL=DYSUM(K,N,L)
     IF(K-KBHX) 916,9156,920
9156 RSUM(K,N,L)=RKNL
     GO TO 9195
916  ZSTEP=ZBH(K+1)-ZBH(K)
     USUM(K,N,L)=UKNL*ZSTEP
     VSUM(K,N,L)=VKNL*ZSTEP
     RSUM(K,N,L)=RKNL*ZSTEP
     DXSUM(K,N,L)=DXKNL*ZSTEP
     DYSUM(K,N,L)=DYKNL*ZSTEP
     DSUM=ZSTEP*DFZ(K,N,L)
     M=K-1
     IF(M) 920,919,918
918  USUM(K,N,L)=USUM(K,N,L)+USUM(M,N,L)
     VSUM(K,N,L)=VSUM(K,N,L)+VSUM(M,N,L)
     RSUM(K,N,L)=RSUM(K,N,L)+RSUM(M,N,L)
     DXSUM(K,N,L)=DXSUM(K,N,L)+DXSUM(M,N,L)
     DYSUM(K,N,L)=DYSUM(K,N,L)+DYSUM(M,N,L)
919  DKAV=DKAV+DSUM
9195 IF(MC(1).LT.1) GO TO 920
     WRITE(ISOUT,5) K,UKNL,VKNL,WFZ(K,N,L),DXKNL,DYKNL,DFZ(K,N,L),RKNL
920  CONTINUE
     IF(MC(1).NE.2) GO TO 9205
     WRITE(ISOUT,6)
     WRITE(ISOUT,7) (K,USUM(K,N,L),VSUM(K,N,L),DXSUM(K,N,L),
    2DYSUM(K,N,L),RSUM(K,N,L),K=1,KBHX)
9205 DKAV=DKAV/ZSPAN
     WRITE(ISOUT,8) L,N,     DKAV
     CALL CNTR(NET,NETSU,N,XG,YG,ICF,JCF,NCF)
     XQ=XG
     YQ=YG
     CALL        NEST(NET,NETSU,XQ,YQ,NDATQ,XL,XR,YL,YU,ICF,JCF,NCF)
     AREAN=(XR-XL)*(YU-YL)
     DAVG(L)=DAVG(L)+DKAV*AREAN
     DO 9210 KL=1,KBHX
9210 WAVG(KL,L)= WAVG(KL,L) + WFZ(KL,N,L)*AREAN
921  CONTINUE
     DAVG(L)=DAVG(L)/AREA
     DO 9215 KL=1,KBHX
9215 WAVG(KL,L)=WAVG(KL,L) / AREA
     WRITE(ISOUT,15) L,        DAVG(L)
     WRITE(ISOUT,16)
     WRITE(ISOUT,17)(K,WAVG(K,L),K=1,KBHX)
922  CONTINUE
     RETURN
     END
```

```
      SUBROUTINE ADVEC(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     1TDEP,CAV,PMAS,PSIZ,ICF,JCF,NCF,KBHF,NDATF,LTIMF,CAVS,WFZ)
C     SEPTEMBER, 1971
C     SUBROUTINE ADVEC TRANSPORTS PARCELS BY SIMPLE ADVECTION PLUS
C     SETTLING. PARCEL TOP AND BASE ARE TRANSPORTED SEPARATELY, AND THE
C     RESULTS ARE SHEARED. THE COMMON VARIABLE ZP IS REDEFINED HEREIN.
C     ZP    - PARCEL CENTER Z COORDINATE BEFORE ADVECTION, ECXEPT AS
C                REDEFINED IN SUB. ADVEC
      COMMON /QPARM/ IPOUT ,IPARIN,NBLK  ,NAT   ,NDELT ,KX    ,KKM
     1,NSEQO ,ICX   ,JCX   ,NCX   ,KBHX  ,NDATX ,LTIMX ,ISIN  ,ISOUT
     2,EDDY  ,FMBEL ,LSTEP ,MC(18),WINT  ,XLLC  ,YLLC  ,THETA ,ZMIN
     3,CSKIP ,MINT  ,ZMAX  ,TIMEX ,DT    ,DZ    ,XP    ,YP    ,ZP
     4,DINCR ,DOWN  ,TP    ,ZLOW  ,DWAF  ,RWAF  ,ROPART,ZUPP  ,VETA
     5,DOPEN ,CROSS ,TIME  ,KKMA1 ,KKX   ,KKXS1 ,KXMIN ,NDATP
      DIMENSION NET(ICF,JCF),NETSU(NCF),ZBH(KBHF),USUM(KBHF,NDATF,LTIMF)
      DIMENSION VSUM(KBHF,NDATF,LTIMF),DXSUM(KBHF,NDATF,LTIMF)
      DIMENSION DYSUM(KBHF,NDATF,LTIMF),TIMUP(LTIMF)
      DIMENSION RSUM(KBHF,NDATF,LTIMF)
      DIMENSION CAVS(KBHF),WFZ(KBHF,NDATF,LTIMF)
      MC3=MC(3)
      EPS=0.1
      NDEP=0
      ZDEP=ZMIN
CHANGE ZP FROM PARCEL CENTER TO PARCEL BASE ALTITUDE.
      ZP=ZLOW
CALCULATE TRANSPORT OF PARCEL BASE.
      IF (    (ZP-ZDEP).LE.EPS) GO TO 1411
      CALL         TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     1NDEP,TDEP,ZDEP,XOL,YOL,ZOL,TOL,SIGXL,SIGYL,ROL,NDATL,ICF,JCF,NCF,
     2KBHF,NDATF,LTIMF,0,CAVS,WFZ)
      GO TO 1412
 1411 TOL=TP
      XOL=XP
      YOL=YP
      ZOL=ZP
      ROL=0.
      SIGXL=RWAF
      SIGYL=RWAF
CHANGE ZP FROM PARCEL BASE TO PARCEL TOP ALTITUDE.
 1412 ZP=ZLOW+DWAF
CALCULATE TRANSPORT OF PARCEL TOP.
      IF( ZP-ZDEP .LE.EPS) GO TO 1414
      CALL         TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     1NDEP,TDEP,ZDEP,XOU,YOU,ZOU,TOU,SIGXU,SIGYU,ROU,NDATU,ICF,JCF,NCF,
     2KBHF,NDATF,LTIMF,0,CAVS,WFZ)
      GO TO 1415
 1414 TOU=TP
      XOU=XP
      YOU=YP
      ZOU=ZP
      ROU=0.
      SIGXU=RWAF
      SIGYU=RWAF
CALCULATE SHEAR OF PARCEL TOP AND BASE RESULTS.
 1415 ZOUTN=(ZOL+ZOU)/2.
      TOUTN=(TOL+TOU)/2.
      IF(ABS(XOU-XOL).GE.1.0E-30) GO TO 1404
      IF(ABS(YOU-YOL).GE.1.0E-30) GO TO 1403
```

```
      ROUTN=0.
      GO TO 1405
 1403 ROUTN=1.57079633
      GO TO 1405
 1404 ROUTN=ATAN((YOU-YOL)/(XOU-XOL))
 1405 R=ROUTN-ROL
      SXL=1./SQRT((COS(R)/SIGXL)**2+(SIN(R)/SIGYL)**2)
      SYL=1./SQRT((SIN(R)/SIGXL)**2+(COS(R)/SIGYL)**2)
      R=ROUTN-ROU
      SXU=1./SQRT((COS(R)/SIGXU)**2+(SIN(R)/SIGYU)**2)
      SYU=1./SQRT((SIN(R)/SIGXU)**2+(COS(R)/SIGYU)**2)
      SXOTN=(SXU+SXL+SQRT((XOU-XOL)**2+(YOU-YOL)**2))/2.
      SYOTN=SQRT(SYU*SYL)
      XOUTN=XOL+(SXOTN-SXL)*COS(ROUTN)
      YOUTN=YOL+(SXOTN-SXL)*SIN(ROUTN)
      CALL DUMPER(XOUTN,YOUTN,ZOUTN,TOUTN,SXOTN,SYOTN,PHAS.PSIZ,ROUTN,0,
     1ISOUT,IPOUT,MC3,NBLK)
      RETURN
      END
```

```
      SUBROUTINE AMBNT(ZCH,DFZ,WFZ,ZHT,W,DIFF,AA,BB,CC,DENOM,E,F,NET,
     1NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,NTRIP,LTIM,ICF,JCF,NCF,
     2KBHF,NDATF,LTIMF,KKF,CAVS)
C     SEPTEMBER 1971
C     SUBROUTINE AMBNT DETERMINES FROM INPUT FALL RATE AND VERTICAL WIND
C     AND TURBULENCE DATA THE COEFFICIENT DATA VECTORS
C     AA(K)=S2*(DIFF(K+1)+DIFF(K))+S1*F(K+1),
C     BB(K)=S2*(DIFF(K+1)+2.*DIFF(K)+DIFF(K-1))+S1*F(K),
C     CC(K)=S2*(DIFF(K)+DIFF(K-1)),
C     DENOM(K)=1+THETA*(BB(K)-CC(K)*E(K-1)),
C     E(K)=THETA*AA(K)/DENOM(K),
C     WHERE
C     S1=DT/DZ,
C     S2=DT/(2.*(DZ)**2)),
C     AND F IS A TEMPORARY WORKING SPACE FOR VERTICAL VELOCITIES.
C     NOTE THAT Q(K) FOR THE NEXT TIME STEP IS GIVEN BY
C     AA(K)*Q(K+1)-BB(K)*Q(K)+CC(K)*Q(K-1).
C     DFKXS1- VERTICAL DIFFUSIVITY AT ALTITUDE INCREMENT KX-1
C     LTIM  - ATMOS. UPDATE INDEX FOR ARRAYS DFZ AND WFZ
C     NTRIP - OPTION CODE FOR HORIZONTAL ADVECTION OF MASS ALOFT
C                POSITIVE IF NDATO IS STORED IN NTRIP
C                NEGATIVE IF NDATO IS TO BE FOUND VIA CALL TO SUB. TRANP
      COMMON /OPARM/ IPOUT ,IPARIN,NBLK  ,NAT   ,NDELT ,KX    ,KKM
     1,NSEOO ,ICX   ,JCX   ,NCX   ,KBHX  ,NDATX ,LTIMX ,ISIN  ,ISOUT
     2,EDDY  ,FMBEL ,LSTEP ,MC(15),WINT  ,XLLC  ,YLLC  ,THETA ,ZMIN
     3,CSKIP ,MINT  ,ZMAX  ,TIMEX ,DT    ,DZ    ,XP    ,YP    ,ZP
     4,DINCR ,DOWN  ,TP    ,ZLOW  ,DWAF  ,PKAF  ,ROPART,ZUPP  ,VETA
     5,DOPEN ,CROSS ,TIME  ,KKMA1 ,KKX   ,KKXS1 ,KKMIN ,NDATP
      COMMON /ODBLE/DFKXS1,EFFLUX,FMAB,PHI,THETO
      DOUBLE PRECISION AA(KKF),BB(KKF),CC(KKF),DENOM(KKF),E(KKF),F(KKF)
      DIMENSION DIFF(KKF),W(KKF),ZHT(KKF),DFZ(KBHF,NDATF,LTIMF)
      DIMENSION WFZ(KBHF,NDATF,LTIMF),TIMUP(LTIMF),ZBH(KBHF),ZCH(KBHF)
      DIMENSION  DXSUM(KBHF,NDATF,LTIMF),DYSUM(KBHF,NDATF,LTIMF)
      DIMENSION   USUM(KBHF,NDATF,LTIMF), VSUM(KBHF,NDATF,LTIMF)
      DIMENSION   RSUM(KBHF,NDATF,LTIMF),NET(ICF,JCF),NETSU(NCF)
      DIMENSION CAVS(KBHF)
      DOUBLE PRECISION DFKXS1,EFFLUX,FMAB,PHI,THETO
      DOUBLE PRECISION S1,S2
      DATA PROGRM/≠AMBNT ≠/
CONSTRUCT F AND DIFF FOR K=0,...,KX
      NDEP=100
      TDEP=TIME
      NDATO=IABS(NTRIP)
COMPUTE KBH AND INITIALIZE
      DIFF(KKM)=0.
      F(KKM)=0.
      ZOLD=ZHT(KKM)
      DOLD=DIFF(KKM)
      FOLD=F(KKM)
      DO 1 K=1,KBHX
      KBH=K
      ZNEW=ZCH(KBH)
      IF(ZOLD.LT.ZNEW) GO TO 2
    1 IF(KBH.EQ.KBHX) CALL ERROR(PROGRM,-1,ISOUT)
    2 IF(NTRIP.LT.-1)
     1CALL        TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     2NDEP,TDEP,ZNEW,XO,YC,ZC,TO,SIGXO,SIGYO,RO,NDATO,ICF,JCF,NCF,KBHF,
     3NDATF,LTIMF,1,CAVS,WFZ)
```

```
      DNEW=DFZ(KBH,NDATO,LTIM)
      FNEW=-WFZ(KBH,NDATO,LTIM)
      DSLOPE=(DNEW-DOLD)/(ZNEW-ZOLD)
      FSLOPE=(FNEW-FOLD)/(ZNEW-ZOLD)
      DO 5 KK=KKMA1,KKX
      ZHTKK=ZHT(KK)
      IF(ZHTKK.LT.ZNEW) GO TO 4
      IF(KBH.LT.KBHX-1) GO TO 3
   40 DIFF(KK)=DNEW
      F(KK)=FNEW
      GO TO 5
    3 KBH=KBH+1
      IF(ZHTKK.GE.ZCH(KBH)) GO TO 30
      DOLD=DNEW
      FOLD=FNEW
      ZOLD=ZNEW
      GO TO 38
   30 KBC=KBH+1
      IF(KBC.LT.KBHX-1) GO TO 32
      KBH=KBC
   31 ZNEW=ZCH(KBH)
      IF(NTRIP.LT.-1)
     1CALL          TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     2NDEP,TDEP,ZNEW,XO,YC,ZO,TO,SIGXO,SIGYO,RO,NDATO,ICF,JCF,NCF,KBHF,
     3NDATF,LTIMF,1,CAVS,WFZ)
      DNEW=DFZ(KBH,NDATO,LTIM)
      FNEW=-WFZ(KBH,NDATO,LTIM)
      GO TO 40
   32 KBHXM1=KBHX-1
      DO 35 K=KBC,KBHXM1
      KBH=K
      IF(ZHTKK.GE.ZCH(KBH)) GO TO 35
      ZOLD=ZCH(KBH-1)
      IF(NTRIP.LT.-1)
     1CALL          TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     2NDEP,TDEP,ZOLD,XO,YC,ZO,TO,SIGXO,SIGYO,RO,NDATO,ICF,JCF,NCF,KBHF,
     3NDATF,LTIMF,1,CAVS,WFZ)
      DOLD=DFZ(KBH-1,NDATO,LTIM)
      FOLD=-WFZ(KBH-1,NDATO,LTIM)
      GO TO 38
   35 CONTINUE
      GO TO 31
   38 ZNEW=ZCH(KBH)
      IF(NTRIP.LT.-1)
     1CALL          TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     2NDEP,TDEP,ZNEW,XO,YC,ZO,TO,SIGXO,SIGYO,RO,NDATO,ICF,JCF,NCF,KBHF,
     3NDATF,LTIMF,1,CAVS,WFZ)
      DNEW=DFZ(KBH,NDATO,LTIM)
      FNEW=-WFZ(KBH,NDATO,LTIM)
      DSLOPE=(DNEW-DOLD)/(ZNEW-ZOLD)
      FSLOPE=(FNEW-FOLD)/(ZNEW-ZOLD)
    4 DIFF(KK)=DOLD+DSLOPE*(ZHTKK-ZOLD)
      F(KK)=FOLD+FSLOPE*(ZHTKK-ZOLD)
    5 CONTINUE
C     CORRECT DIFFUSIVITIES FOR CROSSING TRAJECTORIES EFFECTS
      DO 150 KK=KKM,KKX
  150 DIFF(KK)=DOWN*DIFF(KK)
C     COMBINE STILL-AIR FALL RATES WITH VERTICAL WINDS
```

```
      DO 250 KK=KKM,KKX
  250 F(KK)=F(KK)+H(KK)
COMPUTE E(0) IN ACCORDANCE WITH REFLECTIVITY CONTROL VARIABLE VETA
      E(KKM)=0.
      IF(VETA.LT.0.) GO TO 17
      WREF=F(KKMA1)
      DREF=(DIFF(KKM)+DIFF(KKMA1))/(2.*DZ)
      E(KKM)=(WREF+DREF)/(VETA+DREF)
COMPUTE AA,BB,CC,DENOM, AND E FOR K=1,...,KX-1
   17 S2=DZ
      S1=DT
      S1=S1/S2
      S2=S1/(2.*S2)
      DO 18 KK=KKMA1,KKX
   18 CC(KK)=S2*(DIFF(KK)+DIFF(KK-1))
      DO 19 KK=KKM,KKXS1
   19 AA(KK)=CC(KK+1)+S1*F(KK+1)
      DO 20 KK=KKMA1,KKXS1
      BB(KK)=AA(KK-1)+CC(KK+1)
      DENOM(KK)=1.+THETQ*(BB(KK)-CC(KK)*E(KK-1))
   20 E(KK)=THETQ*AA(KK)/DENOM(KK)
      DFKXS1=DIFF(KKXS1)
C     WRITE(ISOUT,2221)
C2221 FORMAT(≠0CONTENTS OF ARRAYS ZHT, F, DIFF, AA, BB, CC, DENOM, E≠)
C     WRITE(ISOUT,2222) ZHT
C     WRITE(ISOUT,2222) F
C     WRITE(ISOUT,2222) DIFF
C     WRITE(ISOUT,2222) AA
C     WRITE(ISOUT,2222) BB
C     WRITE(ISOUT,2222) CC
C     WRITE(ISOUT,2222) DENOM
C     WRITE(ISOUT,2222) E
C2222 FORMAT(≠0≠(13E10.2))
      RETURN
      END
```

```
      SUBROUTINE SPRVS(NET,NETSU,ZBH,ZCH,TIMUP,USUM,VSUM,DXSUM,DYSUM,
     1RSUM,DFZ,WFZ,DAVG,WAVG,ALT,RHO,ATEMP,AA,BB,CC,DENOM,DIFF,E,F,Q,W,
     2ZHT,ICF,JCF,NCF,KBHF,NDATF,LTIMF,KKF,NATF,CAVS)
C     SEPTEMBER 1971
C     SUBROUTINE SPRVS SUPERVISES DIFFUSIVE AND/OR ADVECTIVE TRANSPORT
C     OF FALLOUT PARCELS LISTED ON TAPE IPARIN. PARCEL PARAMETERS ARE
C     STORED IN ARRAYS XPAR,YPAR,ZPAR,TPAR,PDAM,PSAM,RWFR,DWFR,ZLWF,VWFR
C     ONLY ONE PARCEL IS TRANSPORTED AT A TIME. FOR THIS PARCEL ABOVE
C     ITEMS ARE STORED IN XP,YP,ZP,TP,PSIZ,PMAS,RWAF,DWAF,ZLOW,VWAF.
C     XPAR  - X COORDINATE OF PARCEL CENTER DATA VECTOR (AT TIME TPAR)
C     YPAR  - Y COORDINATE OF PARCEL CENTER DATA VECTOR (AT TIME TPAR)
C     ZPAR  - Z COORDINATE OF PARCEL CENTER DATA VECTOR (AT TIME TPAR)
C     TPAR  - TIME OF DEFINITION OF CLOUD PARCEL DATA VECTOR
C     PDAM  - MIDPOINT OF PARCEL PARTICLE SIZE CLASS DATA VECTOR
C     PSAM  - TOTAL MASS OF PARCEL DATA VECTOR (AT TIME TPAR)
C     RWFR  - RADIUS OF PARCEL AT C. O. M. DATA VECTOR (AT TIME TPAR)
C     DWFR  - PARCEL THICKNESS DATA VECTOR (AT TIME TPAR)
C     ZLWF  - ALTITUDE OF PARCEL BASA DATA VECTOR (AT TIME TPAR)
C     VWFR  - PARCEL VOLUME DATA VECTOR (AT TIME TPAR)
      COMMON /QPARM/ IPOUT ,IPARIN,NBLK  ,NAT   ,NDELT ,KX    ,KKM
     1,NSEQO ,ICX   ,JCX   ,NCX   ,KBHX  ,NDATX ,LTIMX ,ISIN  ,ISOUT
     2,EDDY  ,FMBEL ,LSTEF ,MC(13),WINT  ,XLLC  ,YLLC  ,THETA ,ZMIN
     3,CSKIP ,MINT  ,ZMAX  ,TIMEX ,DT    ,DZ    ,XP    ,YP    ,ZP
     4,DINCR ,DOWN  ,TP    ,ZLOW  ,DWAF  ,RWAF  ,ROPART,ZUPP  ,VETA
     5,DOPEN ,CROSS ,TIME  ,KKMA1 ,KKX   ,KKXS1 ,KXMIN ,NDATP
      COMMON /QDBLE/DFKXS1,EFFLUX,FMAB,PHI,THETC
      DOUBLE PRECISION AA(KKF),BB(KKF),CC(KKF),DENOM(KKF),E(KKF),F(KKF)
      DOUBLE PRECISION Q(KKF)
      DIMENSION ALT(NATF),RHO(NATF),ATEMP(NATF)
      DIMENSION NET(ICF,JCF),NETSU(NCF),ZBH(KBHF),ZCH(KBHF),DIFF(KKF)
      DIMENSION USUM(KBHF,NDATF,LTIMF),VSUM(KBHF,NDATF,LTIMF),W(KKF)
      DIMENSION DXSUM(KBHF,NDATF,LTIMF),DYSUM(KBHF,NDATF,LTIMF)
      DIMENSION DFZ(KBHF,NDATF,LTIMF),ZHT(KKF)
      DIMENSION WFZ(KBHF,NDATF,LTIMF),DAVG(LTIMF),WAVG(KBHF,LTIMF)
      DIMENSION TIMUP(LTIMF),RSUM(KBHF,NDATF,LTIMF)
      DIMENSION XPAR(100),YPAR(100),ZPAR(100),TPAR(100),PDAM(100)
      DIMENSION PSAM(100),RWFR(100),DWFR(100),ZLWF(100),VWFR(100)
      DIMENSION CAVS(KBHF)
      DOUBLE PRECISION DFKXS1,EFFLUX,FMAB,PHI,THETQ
 8014 FORMAT(≠+≠T102,E12.4,I4)
 8015 FORMAT(≠+≠T103,≠AIRBORNE (ADVCN)≠)
 8016 FORMAT(≠ ≠I4,8E12.4)
 8017 FORMAT(≠+≠T103,≠AIRBORNE (DIFFN)≠)
 8018 FORMAT(≠+≠T103,≠ADVECTIVE TRNSPT≠)
 8019 FORMAT(≠+≠T103,≠ IMPACTED WAFER≠)
 8020 FORMAT(≠+≠T103,≠OUTSIDE WINDGRID≠)
 8021 FORMAT(≠0≠36X,≠PARTICLE SIZE CLASS≠E12.4,≠  MICRONS≠)
 8022 FORMAT(≠ ≠22X,≠FALL RATE≠E12.4,≠  METERS/SEC  AT ALTITUDE≠E12.4,≠
     1 METERS≠/22X,≠UPPER LIMIT INPUT ALTITUDE FOR ADV. TRANSP. IS≠E12.4
     2,≠ METERS≠)
 8024 FORMAT(≠0≠T2,≠NSEQ≠T11,≠XP≠T23,≠YP≠T35,≠ZP≠T47,≠TP≠T58,≠PMAS≠T70,≠
     1RWAF≠T82,≠ZLOW≠T94,≠DWAF≠T107,≠DZ≠T115,≠KX≠/)
 8025 FORMAT(≠0NEGATIVE DEPOSIT. WAFER NO.≠I4,≠ AT TIME≠E12.4,≠. VARIABL
     5ES EFFLUX,FMAB,DEP ≠2D12.4,E12.4/)
      DATA  PROGRM/≠SPRVS ≠/
      JF=100
      KKMA1=KKM+1
      THETQ=THETA
```

```
      PHI=1.-THETQ
      KXBF=0
      MC3=MC(3)
      NSEQ=0
      PSZBE=-2.0
C     COMPUTE OVERALL AVERAGE VERTICAL VELOCITY FOR THE FIRST UPDATE
      WAVGK=0.0
      KBHM1=KBHX-1
      DO 50 K=1,KBHM1
   50 WAVGK=WAVGK + WAVG(K,1)*(ZBH(K+1) - ZBH(K))
      WAVGK=WAVGK/(ZBH(KBHX)-ZBH(1))
C     COMPUTE TIMEX MARGIN FACTOR FOR ADVECTIVE TRANSPORT AIRBORNE TEST
      IF(NDATX-1)70,70,60
   60 SLOP=1.1
      GO TO 80
   70 SLOP=1.0
C     COMPUTE MINIMUM SMALL ALTITUDE INCREMENT DZMIN
   80 DZMIN=(ZMAX-ZMIN)/KX
C     CUE IPARIN TAPE AT BEGINNING OF INPUT PARCEL BLOCK
  100 READ(IPARIN) NP
      IF(NP.LE.0) GO TO 806
      IF(NP.GT.JF) CALL ERROR(PROGRM,-100,ISOUT)
C     COPY IN A BLOCK OF INPUT PARCEL PARAMETERS FROM TAPE IPARIN
      READ(IPARIN) (XPAR(J),YPAR(J),ZPAR(J),TPAR(J),PDAM(J),PSAM(J),
     1RWFR(J),DWFR(J),ZLWF(J),VWFR(J),J=1,NP)
C     COMMENCE PROCESSING BLOCK OF INPUT PARCELS ONE AT A TIME
      DO 1000 J=1,NP
      NSEQ=NSEQ+1
      IF(NSEQ.LT.NSEQO) GO TO 1000
      XP=XPAR(J)
      YP=YPAR(J)
      ZP=ZPAR(J)
      TP=TPAR(J)
      PSIZ=1.0E6*PDAM(J)
      PMAS=PSAM(J)
      RWAF=RWFR(J)/2.
      DWAF=DWFR(J)
      ZLOW=ZLWF(J)
      VWAF=VWFR(J)
C     CHECK FOR NEW PARTICLE SIZE CLASS
      IF(ABS((PSIZ-PSZBE)/PSIZ).LE.1.0E-10) GO TO 103
      WRITE(ISOUT,8021) PSIZ
C     COMPUTE MID-ATMOSPHERE FALL RATE FAV FOR NEW PARTICLE SIZE CLASS
      H=(ZMIN+ZMAX)/2.
      CALL TRPL(H,NAT,ALT,RHO,DEN)
      CALL TRPL(H,NAT,ALT,ATEMP,VIS)
      CALL  FALRT(PSIZ,ROPART,H,DEN,VIS,FAV,ISOUT)
      FAV=FAV-WAVGK
C     COMPUTE UPPER LIMIT ALTITUDE FOR ADVECTIVE TRANSPORT OF THIS SIZE PART.
      CALL CALIB(ZBH,KBHX,ZMIN,-1,KBHZ)
      CALL TRPL(ZBH(KBHZ),NAT,ALT,ATEMP,VIS)
      CALL TRPL(ZBH(KBHZ),NAT,ALT,RHO,DEN)
      CALLFALRT(PSIZ,ROPART,ZBH(KBHZ),DEN,VIS,CAV,ISOUT)
      TDEP=TP + (ZBH(KBHZ) - ZMIN) /(CAV - WAVG(KBHZ-1,1))
      KBHM1=KBHX-1
      DO 1001 IZ=KBHZ,KBHM1
      CALL TRPL(ZCH( IZ ),NAT,ALT,ATEMP,VIS)
      CALL TRPL(ZCH( IZ ),NAT,ALT,RHO,DEN)
```

```
      CALLFALRT(PSIZ,ROPART,ZCH( IZ ),DEN,VIS,CAV,ISOUT)
      TDEP=TDEP + (ZBH(IZ+1) - ZBH(IZ))/(CAV- WAVG(IZ,1))
      IF(TDEP.GT.SLOP*TIMEX) GO TO 1002
 1001 CONTINUE
      ZLIM=5.0E4
      GO TO 1003
 1002 ZLIM=ZBH(IZ+1)
 1003 WRITE(ISOUT,8022) FAV,H,ZLIM
      WRITE(ISOUT,8024)
COMPUTE PARTICLE FALL RATE TABLE FOR EACH ATMOSPHERIC STRATUM WHEN
C     A NEW PARTICLE SIZE IS ENCOUNTERED
      DO 101 KKZ=1,KBHX
      CALL TRPL(ZCH(KKZ),NAT,ALT,ATEMP,VIS)
      CALL TRPL(ZCH(KKZ),NAT,ALT,RHO,DEN)
 101  CALL FALRT(PSIZ,ROPART,ZCH(KKZ),DEN,VIS,CAVS(KKZ),ISOUT)
COMPUTE DIFFUSIVITY CORRECTIONS FOR NEW PARTICLE SIZE CLASS
      DOWN=(FAV*EDDY)**2
      CROSS=1./SQRT(1.+4.*DOWN)
      DOWN=1./SQRT(1.+DOWN)
      PSZBE=PSIZ
 103  WRITE(ISOUT,8016) NSEQ,XP,YP,ZP,TP,PMAS,RWAF,ZLOW,DWAF
CANCEL PROCESSING OF PARCEL IF IT HAS ALREADY IMPACTED
      IF (IFIX(DWAF).GT.0) GO TO 1200
      WRITE(ISOUT,8019)
      CALL        DUMPER(XP,YP,ZP,TP, RWAF, RWAF, PMAS,PSIZ,0.,0,
     1ISOUT,IPOUT,MC3,NBLK)
      GO TO 1000
COMPUTE INDEX OF MESH OR SUB-MESH CONTAINING PARCEL CENTER POSITION
 1200 CALL NEST(NET,NETSU,XP,YP,NDATP,XL,XR,YL,YU,ICF,JCF,NCF)
CANCEL PROCESSING OF PARCEL IF IT IS INPUT OUTSIDE ATMOS.
      IF(NDATP.GT.0) GO TO 1248
      WRITE(ISOUT,8020)
      GO TO 1000
COMPUTE AVERAGE FALL RATE CAV
 1248 ZUPP=ZLOW+DWAF
      ZLO=ZLOW-ZMIN
      ZUP=ZUPP-ZMIN
      H=ZMIN+ZUP/2.
      CALL TRPL(H,NAT,ALT,RHO,DEN)
      CALL TRPL(H,NAT,ALT,ATEMP,VIS)
      CALL  FALRT(PSIZ,ROPART,H,DEN,VIS,CAV,ISOUT)
CANCEL PROCESSING OF PARCEL IF IT WILL REMAIN AIRBORNE BY DIFFUSION
      DAV=DOWN*DAVG(1)
      CAV=CAV-WAVGK
      TFLY=TIMEX-TP
      IF(TFLY.LE.0.) GO TO 1249
      CALL ESTM(ZUP,CAV,DAV,TFLY,PUP)
      CALL ESTM(ZLO,CAV,DAV,TFLY,PLO)
      PDEST=1.0-(PUP-PLO)/(ZUP-ZLO)
      IF(CSKIP.LE.PDEST) GO TO 1250
 1249 WRITE(ISOUT,8017)
      GO TO 1000
COMPUTE TRANSPORT BY ADVECTION IF TRUNCATION ERROR IS EXCEEDED
 1250 IF(DZMIN.LE.0.2*DAV/FAV) GO TO 1500
CANCEL PROCESSING OF PARCEL IF IT WILL REMAIN AIRBORNE BY ADVECTION
      IF(ZLOW.LT.ZLIM) GO TO 1409
      WRITE(ISOUT,8015)
      GO TO 1000
```

```
 1409 WRITE(ISOUT,8018)
      CALL       ADVEC(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     1TDEP,CAV,PMAS,PSIZ,ICF,JCF,NCF,KBHF,NDATF,LTIMF,CAVS,WFZ)
      GO TO 1000
C     COMPUTE SMALL ALTITUDE INCREMENT DZ
 1500 DZMAX=DWAF
 1501 IF(DZMAX.GE.DZMIN) GO TO 1502
      DZMAX=2.0*DZMAX
      GO TO 1501
 1502 DZ=0.2*DAV/FAV
      IF(DZ.GT.DZMAX) DZ=DZMAX
      KX=(ZMAX-ZMIN)/DZ+1.
      IF(KX.LT.KXMIN) KX=KXMIN
      DZ=(ZMAX-ZMIN)/KX
      WRITE(ISOUT,8014) DZ,KX
C     COMPUTE ALTITUDE INCREMENT AND FALL RATE DATA VECTORS
      IF(KX.EQ.KXBE) GO TO 1305
      KKX=KKM+KX
      KKXS1=KKX-1
      DO 1304 KK=1,KKX
      ZHT(KK)=ZMIN+DZ*(KK-KKM)
      CALL TRPL(ZHT(KK),NAT,ALT,RHO,DEN)
      CALL TRPL(ZHT(KK),NAT,ALT,ATEMP,VIS)
 1304 CALL  FALRT(PSIZ,ROFAPT,ZHT(KK),DEN,VIS,W(KK),ISOUT)
      KXBE=KX
C     COMPOSE INITIAL CONCENTRATION DATA VECTOR
 1305 CALL        CONC(ZHT,Q,KKF)
C     CANCEL PROCESSING OF WAFER IF INITIAL AIRBORNE MASS IS INADEQUATE
      IF(FMAB.LT.1) GO TO 1000
      DEPBE=0.
      NTRIP=NDATP
C     COMPUTE FIRST DEPOSIT INCREMENT TIME INTERVAL TDELT
      TDELT=TFLY/NDELT
      TLARG=TDELT
      TAE=4.*ZUP/(3.*CAV)
      IF(TAE.LT.TFLY) TDELT=TAE/NDELT
      TSMAL=MINT*DT
      IF(TDELT.LT.TSMAL) TDELT=TSMAL
C     COMMENCE DEPOSIT TIME LOOP
      TIME=TP
      TPAUS=TIME
      LTIM=-1
      NPASS=1
      XOBE=XP
      YOBE=YP
    1 LSTEP=TDELT/DT +1.
      TDELT=LSTEP*DT
  135 TPAUS=TPAUS+TDELT
C     COMPUTE DATA SET TIME INDEX LTIM
      CALL CALIB(TIMUP,LTIMX,TIME,+1,LTIMA)
      IF(LTIM.NE.LTIMA) GO TO 3
      IF(IABS(NTRIP)-1) 31,32,31
    3 LTIM=LTIMA
C     COMPOSE CONCENTRATION COEFFICIENT DATA ARRAYS
   31 CALL        AMBNT(ZCH,DFZ,WFZ,ZHT,W,DIFF,AA,BB,CC,DENOM,E,F,NET,
     1NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,NTRIP,LTIM,ICF,JCF,NCF,
     2KBHF,NDATF,LTIMF,KKF,CAVS)
      NTRIP=-NDATX
```

```
COMPUTE NUMBER OF ITERATIONS LSTEP OF SOLUTION TO VERTICAL DIFFUSION
C      DIFFERENCE EQUATION FOR THIS DEPOSIT INCREMENT TIME INTERVAL
   32 IF(LTIM.GE.LTIMX) GO TO 138
      IF(TPAUS.LT.TIMUP(LTIM+1)) GO TO 140
      TPAUS=TIMUP(LTIM+1)
      TDELT=TPAUS-TIME
      LSTEP=TDELT/DT+1.
      TDELT=LSTEP*DT
  138 IF(TPAUS.LT.TIMEX) GO TO 140
      TPAUS=TIMEX
      TDELT=TPAUS-TIME
      LSTEP=TDELT/DT+1.
      TDELT=LSTEP*DT
      IF(LSTEP.LE.0) GO TO 1000
  140 IF(NPASS.NE.1) GO TO 4
      NPASS=2
      TDELTP=TDELT
COMPUTE DEPOSIT INCREMENT FRACTIONAL MASS DEP
    4 CALL         DIFFF(Q,AA,BB,CC,DENOM,E,F,DEP,KKF)
CHECK DEPOSIT INCREMENT FRACTIONAL MASS DEP AGAINST DOPEN
      IF(DEP.GE.-DOPEN*TDELT) GO TO 5
      WRITE(ISOUT,3025) NSEQ,TIME,EFFLUX,FMAB,DEP
    5 IF(DEP.GE.DOPEN*TDELT) GO TO 7
CHECK CUMULATIVE AIRBORNE FRACTIONAL MASS FMAB AGAINST FMBEL
      IF(SNGL(FMAB).GT.FMBEL) GO TO 135
      GO TO 1000
COMPARE RATE OF CHANGE OF DEPOSITION RATE DPDP WITH DINCR AND ADJUST
C      NEXT TDELT
    7 DPDP=((DEP/TDELT)-(DEPBE/TDELTB))/TDELT
      IF(DPDP.LT.DINCR) GO TO 10
      TDELTB=TDELT
      TDELT=TDELT/2.
      IF(TDELT.LT.TSMAL) TDELT=TSMAL
      GO TO 13
   10 IF(DPDP.GT.DINCR) GO TO 13
      TDELTB=TDELT
      TDELT=2.*TDELT
      IF(TDELT.GT.TLARG) TDELT=TLARG
COMPUTE DEPOSIT INCREMENT MASS PMDEP
   13 PMDEP=PMAS*DEP
      DEPBE=DEP
      NDEP=0
      ZDEP=ZMIN
      TDEP=TIME
COMPUTE DEPOSIT INCREMENT POSITION (XO,YO,ZO) AND HORIZONTAL DISPERSION
C      PARAMETERS (SIGXO,SIGYO,RO) AT TIME=TO
      CALL         TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,PSUM,
     1NDEP,TDEP,ZDEP,XO,YO,ZO,TO,SIGXO,SIGYO,RO,NDATO,ICF,JCF,NCF,KBHF,
     2NDATF,LTIMF,1,CAVS,WF7)
CONTINUE ON TO NEXT WAFER IF THIS ONE LIES OUTSIDE WINDFIELD
      IF(NDATO.LE.0) GO TO 1000
COLLECT FINAL RESULTS FOR THIS WAFER AND STORE IN BUFFER DATA VECTORS
      XH=(XO+XOBE)/2.
      YH=(YO+YOBE)/2.
      CALL         DUMPER(XH,YH,ZO,TO,SIGXO,SIGYO,PMDEP,PSIZ,RO,0,
     1ISOUT,IPOUT,MC3,NBLK)
      XOBE=XO
      YOBE=YO
```

```
      IF(TIME+DT.LT.TIMEX) GO TO 1                                      SPPVS291
 1000 CONTINUE                                                          SPPVS292
      GO TO 100                                                         SPPVS293
COPY OUT BUFFER DATA VECTCRS. WAFER PROCESSING HAS BEEN COMPLETED       SPRVS294
  806 CALL        DUMPER(0.,0.,0.,0.,    0.,    0.,    0.,   0.,0.,999, SPRVS295
     1ISOUT,IPOUT,MC3,NBLK)                                             SPPVS296
      CALL        DUMPER(0.,0.,0.,0.,    0.,    0.,    0.,   0.,0.,999, SPPVS297
     1ISOUT,IPOUT,MC3,NBLK)                                             SPPVS298
      REWIND IPARIN                                                     SPRVS299
      END FILE IPOUT                                                    SPPVS300
      REWIND IPOUT                                                      SPPVS301
      RETURN                                                            SPPVS302
      END                                                               SPPVS303
```

```
      SUBROUTINE TRANP(NET,NETSU,ZBH,TIMUP,USUM,VSUM,DXSUM,DYSUM,RSUM,
     1NDEP,TDEP,ZDEP,XO,YC,ZC,TO,SIGXO,SIGYO,RO,NDATO,ICF,JCF,NCF,KBHF,
     2NDATF,LTIMF,KRIP,CAVS,WFZ)
C     SEPTEMBER 1971
C     SUBROUTINE TRANP DETERMINES (AT AN INPUT TERMINAL ALTITUDE PLANE)
C     THE WAFER HORIZONTAL CENTER POSITION AND DISPERSION PARAMETERS FOR
C     AN INPUT TRANSPORT FLIGHT TIME
C     CAVS   - PARTICLE FALL RATE TABULATED FOR EACH ALTITUDE STRATUM
C     DXSUM - TURBULENCE X COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     DYSUM - TURBULENCE Y COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     NET    - HORIZONTAL SPACE CONTROL NET MESH 2-DIM. ARRAY
C     NETSU - HORIZONTAL SPACE CONTROL NET SUB-MESH DATA VECTOR
C     RSUM   - WIND HEADING ORIENTATION ANGLE (WEIGHTED SUM) 3-DIM. ARRAY
C     USUM   - WIND X COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     VSUM   - WIND Y COMPONENT (WEIGHTED SUM) 3-DIM. DATA ARRAY
C     ZBH    - ATMOSPHERE STRATA BASE-ALTITUDE DATA VECTOR
C     MODE   - COMPUTATION MODE SWITCH
C                 0 RAPID COMPUTATION
C                 1 LAYERWISE COMPUTATION
C     NDEP   - OPTION CONTROL VARIABLE
C                  ZERO IF SIGXO AND SIGYO ARE TO BE COMPUTED
C                  NON-ZERO IF SIGXO AND SIGYO ARE NOT TO BE COMPUTED AND
C                  IF NDATO IS TO BE POSITIVE ALWAYS
C     TDEP   - ADVECTIVE TRANSPORT TIME INTERVAL
C     KRIP    - CONTROL VARIABLE
C                 0 FOR ADVECTIVE TRANSPORT
C                 1 FOR DIFFUSIVE TRANSPORT
C     WFZ    - VERTICAL WIND FIELD
C     ZDEP   - ADVECTIVE TRANSPORT TERMINAL ALTITUDE
C     TO     - TIME AFTER PARCEL ADVECTION
C     XO     - PARCEL CENTER X COORDINATE AFTER ADVECTION
C     YO     - PARCEL CENTER Y COORDINATE AFTER ADVECTION
C     ZO     - PARCEL CENTER Z COORDINATE AFTER ADVECTION
C     SIGXO - PARCEL MASS HOR. STAND. DEV. DOWNWIND AFTER ADVECTION
C     SIGYO - PARCEL MASS HOR. STAND. DEV. CROSSWIND AFTER ADVECTION
C     NDATO - HORIZONTAL INDEX OF LATTICE CELL CONTAINING POINT (XC,YC)
C     RO     - WIND HEADING ORIENTATION ANGLE AFTER ADVECTION
      COMMON /OPARM/ IFOUT ,IPARIN,NBLK  ,NAT   ,NDELT ,KX    ,KKM
     1,MSEQC ,ICX   ,JCX   ,NCX   ,KBHX  ,NDATX ,LTIMX ,ISIN  ,ISCUT
     2,EDDY  ,FMDEL ,LSTEP ,MC(13),WINT  ,XLLC  ,YLLC  ,THETA ,ZMIN
     3,CSKIP ,MINT  ,ZMAX  ,TIMEX ,DT    ,DZ    ,XP    ,YP    ,ZP
     4,DINCP ,DCWN  ,TP    ,ZLOW  ,DKAF  ,RWAF  ,ROPART,ZUPP  ,VETA
     5,DOPEN ,CROSS ,TIME  ,KKMA1 ,KKX   ,KKXS1 ,KXMIN ,NDATF
      DIMENSION NET(ICF,JCF),NETSU(NCF),ZBH(KBHF),USUM(KBHF,NDATF,LTIMF)
      DIMENSION VSUM(KBHF,NDATF,LTIMF),DXSUM(KBHF,NDATF,LTIMF)
      DIMENSION DYSUM(KBHF,NDATF,LTIMF),TIMUP(LTIMF)
      DIMENSION CAVS(KBHF)
      DIMENSION RSUM(KBHF,NDATF,LTIMF)
      DIMENSION WFZ(KBHF,NDATF,LTIMF)
      DATA PROGRM/*TRANP */
    2 FORMAT(* TIME=*E12.4,*. ALT=*E12.4,*. X-POS=*E12.4,*. Y-POS=*E12.4
     2,*. CELL=*I6,* REACHED*)
    3 FORMAT(* TIME=*E12.4,*. ALT=*E12.4,*. X-POS=*E12.4,*. Y-POS=*E12.4
     3,*. CELL=*I6,* ATTEMPTED*)
    4 FORMAT(*0WAFER WITH INITIAL CONFIGURATION XP,YP,ZP,TP *4E12.4/*REQ
     1UIRED CHANNELLING AT CONFIGURATION XC,YC,ZC,TC *4E12.4)
      EPSILO=.0005
      EPS=EPSILO*WINT
```

```
      EPST=EPSILO*TDEP
      EPS7=.1
      XO=XP
      YO=YP
      ZO=ZP
      TO=TP
      SIGXO=0.
      SIGYO=0.
      RO=0.
      NDATO=NDATP
      NDTC1=0
      NDTO1=0
      KBHC1=0
      KBHO1=0
 1000 CONTINUE
      KAY=-1
      IF (KRIP.EQ.1) GO TO 50
CALCULATE FALL VELOCITY FOR ADVECTION ON BASIS OF LOCAL
C     FALL RATE AND WIND FIELD  AND STORE IN WBAR
      CALL CALIB(ZBH,KBHX,ZO,KAY,KBHZ)
      CALL CALIB(TIMUP,LTIMX,TO,1,LTIM)
      WBAR=WFZ(KBHZ-1,NDATO,LTIM)-CAYS(KBHZ-1)
      IF (WBAR) 112,111,110
COMPUTE THE VERTICAL PSEUDO-VELOCITY WBAR AND STORE ITS SIGN IN KAY
C     FOR THE DIFFUSIVE SETTLING CASE
   50 WBAR=(ZDEP-ZP)/(TDEP-TP)
      IF(WBAR) 112,111,110
  110 KAY=KAY+1
  111 KAY=KAY+1
  112 CONTINUE
CALIBRATION OF ZDEP AGAINST ZBH YIELDS TERMINAL ZBH-PLANE
      IF (KRIP.EQ.1) CALL CALIB(ZBH,KBHX,ZDEP,KAY,KBH)
CONSIDER KAY=0 CASE INDEPENDENTLY
  200 IF(KAY.NE.0) GO TO 206
      IF (KRIP.EQ.1) GO TO 205
C     IN THE ADVECTIVE TRANSPORT CASE WHENEVER THE ACTUAL FALL RATE
C     IS ZERO THEN SET THE DEPOSITION TIME INCREMENT EQUAL TO THE
C     TIME LEFT BEFORE  THE WIND FIELD IS UPDATED
      TSEG=TIMUP(LTIM+1)-TO
      KBHC=KBHZ+1
      KBHO=KBHZ
      GO TO 300
  205 TSEG=TDEP-TO
      MODE=1
      KBHC=KBH
      KBHO=KBHC-1
      GO TO 300
CALIBRATION OF ZO AGAINST ZBH YIELDS CURRENT ZBH-PLANE
  206 CALL CALIB(ZBH,KBHX,ZO,-KAY,KTRY)
      CALL CALIB(ZBH,KBHX,ZO,+KAY,KBHC)
      IF ((KRIP.EQ.0).OR.(KBHO.NE.KTRY)) GO TO 213
CONSIDER EXCURSION TO TERMINAL ZBH-PLANE
      KBHC=KBH
      ZEST=ZBH(KBHC)
      MODE=0
      IF(KAY*(KBHO-KBHC)+1) 221,213,218
CONSIDER EXCURSION BETWEEN ADJACENT ZBH-PLANES
  213 KBHC=KBHC+KAY
```

```
      ZEST=ZBH(KBHC)
      MODE=1
      IF((KBHC.LE.KBHX).AND.(KAY*ZEST.LE.KAY*ZDEP)) GO TO 221
CONSIDER EXCURSION TO TERMINAL ZDEP-PLANE
  218 KBHC=KBHO+KAY
      ZEST=ZDEP
      MODE=1
  221 TSEG=(ZEST-ZO)/WBAR
CHECK IF UPDATE TIME BOUNDARY WILL BE CROSSED
  300 TC=TO+TSEG
      CALL CALIB(TIMUP,LTIMX,TO,1,LTIM)
      IF((LTIM.LT.LTIMX).AND.(TIMUP(LTIM+1).LE.TC))TSEG=TIMUP(LTIM+1)-TO
COMPUTE AVERAGE HORIZONTAL VELOCITIES UBAR AND VBAR
      KBHA=KBHO
      KBHB=KBHC
      IF(KAY.LT.0) GO TO 405
      KBHA=KBHC
      KBHB=KBHO
  405 CALL GETDA( USUM,ZBH,KBHA,KBHB,NDATO,LTIM, UBAR,KBHF,NDATF,LTIMF)
      CALL GETDA( VSUM,ZBH,KBHA,KBHB,NDATO,LTIM, VBAR,KBHF,NDATF,LTIMF)
  407 IF(NDEP.NE.0) GO TO 412
COMPUTE AVERAGE HORIZONTAL DISPERSION AND WIND ORIENTATION ANGLE
      CALL GETDA(DXSUM,ZBH,KBHA,KBHB,NDATO,LTIM,DXBAR,KBHF,NDATF,LTIMF)
      CALL GETDA(DYSUM,ZBH,KBHA,KBHB,NDATO,LTIM,DYBAR,KBHF,NDATF,LTIMF)
      CALL GETDA( RSUM,ZBH,KBHA,KBHB,NDATO,LTIM, RBAR,KBHF,NDATF,LTIMF)
      RC=RO+RBAR
      SIGXC=SIGXO+DXBAR*TSEG
      SIGYC=SIGYO+DYBAR*TSEG
COMPUTE CURRENT POSITION AND TIME (XC,YC,ZC,TC)
  412 TC=TO+TSEG
      ZC=ZO+WBAR*TSEG
      XC=XO+UBAR*TSEG
      YC=YO+VBAR*TSEG
      CALL NEST(NET,NETSU,XC,YC,NDATC,XL,XR,YL,YU,ICF,JCF,NCF)
      IF(MC(4).EQ.1) WRITE(ISOUT,3) TC,ZC,XC,YC,NDATC
COMPARE CURRENT MESH INDEX NDATC WITH PREVIOUS MESH INDEX NDATO
      IF(NDATC.EQ.NDATO) GO TO 700
COMPUTE INTERPOLATED POINT
      XT=XC
      YT=YC
      ZT=ZC
      IF (NODE.EQ.0) GO TO 213
      CALL BCUN(NET,NETSU,XT,YT,XO,YO,XC,YC,ICF,JCF,NCF)
      ZC=SQRT(((XT-XC)**2+(YT-YC)**2)/((XT-XO)**2+(YT-YO)**2))
      ZC=ZT+ZC*(ZO-ZT)
      IF(ABS(WBAR).LE.1.0E-30) GO TO 510
      TSEG=(ZC-ZO)/WBAR
      GO TO 518
  510 IF(ABS(UBAR).LE.1.0E-30) GO TO 513
      TSEG=(XC-XO)/UBAR
      GO TO 518
  513 IF(ABS(VBAR).LE.1.0E-30) GO TO 516
      TSEG=(YC-YO)/VBAR
      GO TO 518
  516 CALL ERROR(PROGPM,516,ISOUT)
      RETURN
  518 IF(NDEP.NE.0) GO TO 521
      RC=RO+RBAR
```

```
      SIGXC=SIGXO+DXBAR*TSEG
      SIGYC=SIGYO+DYBAR*TSEG
  521 TC=TO+TSEG
      CALL NEST(NET,NETSU,XC,YC,NDATC,XL,XR,YL,YU,ICF,JCF,NCF)
CHECK IF PARCEL CENTER POSITION IS OSCILLATING
      IF((KBHO1.NE.KBHO).OR.(KBHC1.NE.KBHC).OR.(NDTC1.NE.NDATC).OR.
     1(NDTO1.NE.NDATO)) GO TO 626
      IF(MC(4).EQ.1) WRITE(ISOUT,4) XP,YP,ZP,TP,XC,YC,ZC,TC
      CALL CNTR(NET,NETSU,NDATO,XG,YG,ICF,JCF,NCF)
      XQ=XG
      YQ=YG
      CALL NEST(NET,NETSU,XQ,YQ,NDATQ,XLO,XRO,YLO,YUO,ICF,JCF,NCF)
CLEAR STORED MESH AND STRATUM INDICES
      NDTC1=0
      NDTO1=0
      KBHC1=0
      KBHO1=0
CHANNEL WAFER CENTER POSITION ALONG APPROPRIATE CELL BOUNDARY
      SPE=2.*EPS
      IF((ABS(XLO-XR).GT.SPE).AND.(ABS(XRO-XL).GT.SPE)) GO TO 616
      UBAR=0.
      CALL GETDA( VSUM,ZBH,KBHA,KBHB,NDATO,LTIM,VBARC,KBHF,NDATF,LTIMF)
      IF(ABS(VBARC).LE.ABS(VBAR)) GO TO 407
      VBAR=VBARC
      NDATO=NDATC
      GO TO 407
  616 IF((ABS(YLO-YU).GT.SPE).AND.(ABS(YUO-YL).GT.SPE))
     1CALL ERROR(PROGPM,616,ISOUT)
      VBAR=0.
      CALL GETDA( USUM,ZBH,KBHA,KBHB,NDATO,LTIM,UBARC,KBHF,NDATF,LTIMF)
      IF(ABS(UBARC).LE.ABS(UBAR)) GO TO 407
      UBAR=UBARC
      NDATO=NDATC
      GO TO 407
COMMIT PREVIOUS AND CURRENT MESH AND STRATUM INDICES TO STORAGE
  626 NDTC1=NDATO
      NDTO1=NDATC
      KBHC1=KBHC
      KBHO1=KBHO
CONVERT XO,YO,ZO,TO,SIGXO,SIGYO, AND NDATC TO CURRENT VALUES
  700 ZO=ZC
      XO=XC
      YO=YC
      TO=TC
      NDATO=NDATC
      IF(MC(4).EQ.1) WRITE(ISOUT,2) TO,ZC,XO,YO,NDATO
      IF(NDEP.NE.0) GO TO 708
      SIGXO=SIGXC
      SIGYO=SIGYC
      RO=RC
CHECK IF CURRENT POSITION IS OUTSIDE ATMOSPHERE
  708 IF(NDATO.LE.0) GO TO 710
      IF (KRIP.EQ.1) GO TO 709
C     IF DEPOSITION PLANE IS REACHED OR TRANSPORT TIME LIMIT IS EXCEEDED
C     EXIT FROM TRANP, OTHERWISE RETURN TO TOP
      IF ((   (ZO-ZDEP).LE.EPS2).OR.(  (TIMSX-TO).LE.EPST)) GO TO 720
      GO TO 1000
  709 CONTINUE
```

```
CHECK IF TOTAL FLIGHT TIME HAS BEEN EXHAUSTED                          TRANP233
      IF(TO+EPST-TDEP) 200,720,720                                     TRANP234
CARRY PARCEL CENTER BACK INTO ATMOS. IF NDEP IS NOT EQUAL TO ZERO      TRANP235
  710 IF(NDEP.EQ.0) GO TO 720                                          TRANP236
      XO=XO-2.*EPS                                                     TRANP237
      YO=YO-2.*EPS                                                     TRANP238
      CALL NEST(NET,NETSU,XO,YO,NDATO,XL,YR,YL,YU,ICF,JCF,NCF)         TRANP239
      IF(NDATO.GT.0) GO TO 720                                         TRANP240
  714 XO=XO+4.*EPS                                                     TRANP241
      YO=YO+4.*EPS                                                     TRANP242
      CALL NEST(NET,NETSU,XO,YO,NDATO,XL,YR,YL,YU,ICF,JCF,NCF)         TRANP243
      IF(NDATO.LE.0) CALL ERROR(PROGRM,720,ISOUT)                      TRANP244
COMPUTE HORIZ. DISPERSION IF NDEP IS NOT EQUAL TO ZERO                 TRANP245
  720 IF(NDEP.NE.0) RETURN                                             TRANP246
      R2=PWAF**2                                                       TRANP247
      IF(MC(10).EQ.1) GO TO 721                                        TRANP248
      SIGXC=2.*DOWN*SIGXC                                              TRANP249
      SIGYC=2.*CROSS*SIGYC                                             TRANP250
      GO TO 722                                                        TRANP251
  721 TRIP=TC-TP                                                       TRANP252
      DSPRTX=SIGXO/TRIP                                                TRANP253
      TONEX=4.*((R2/DSPRTX)**(1./3.))                                  TRANP254
      IF(TRIP.LE.TONEX) SIGXC=DSPRTX*TONEX*(TRIP**2)/3.                TRANP255
      IF(TRIP.GT.TONEX) SIGXC=DSPRTX*(TRIP**3)/3.                      TRANP256
      SIGXO=SIGXO*(DOWN**(3./2.))                                      TRANP257
      DSPRTY=SIGYO/TRIP                                                TRANP258
      TONEY=4.*((R2/DSPRTY)**(1./3.))                                  TRANP259
      IF(TRIP.LE.TONEY) SIGYC=DSPRTY*TONEY*(TRIP**2)/3.                TRANP260
      IF(TRIP.GT.TONEY) SIGYC=DSPRTY*(TRIP**3)/3.                      TRANP261
      SIGYO=SIGYO*(CROSS**(3./2.))                                     TRANP262
  722 SIGXO=SQRT(R2+SIGXO)                                             TRANP263
      SIGYO=SQRT(R2+SIGYO)                                             TRANP264
      RETURN                                                           TRANP265
      END                                                              TRANP266
```